

DETAILED ACTION

- A. This action is in response to the following communications: Amendment filed: 12/29/2009. This action is made **Final**.
- B. Claims 1-9 and 11-21 remain pending.

Specification

1. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: As for claims 1-7, the term "computer-readable medium" lacks antecedent basis by not being defined within the specification. Applicant is advised to consider an alternative such as "program media".

Allowable Subject Matter

2. Claims 6, 14 and 20 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Specifically allowable subject matter consist of: code for assigning a dynamic priority to each conflicting region having a common overlapping region, the dynamic priority based on the projected amount of time expended to render each conflicting region; and code for selecting the agent controlling the conflicting region having the highest priority to retain control over the overlapping region.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-5, 7-9, 11-13, 15-19 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mutschler et al. (US Pat. 5,940,075), herein referred to as "Mutschler" in view of Galvin, Michael A. et al. (US pat. 7,281,236 B1), herein referred to as "Galvin".

Summary: Mutschler provides a system that enables a client user to access a character based interface (legacy system) by means of transformation of the older interface into a web served graphical use interface.

As for independent claims 1, 8 and 15, Mutschler teaches a computer-readable medium having stored and corresponding computer system and method for recognizing computer program code that when executed by a computer cause a computer system to recognize a character based user interface having a plurality of host component types and to transform the character based user interface to a web enabled user interface, the medium having code to control the computer, the medium/system/method containing code for/steps for (col.3, lines 35-60): scanning the character based user interface for a

plurality of agents (col.6,lines 24-32, 54-64); in each agent for determining the existence of a different host component type unique to the agent (col.9,lines 23-47; col.6,lines 24-32, 54-64); defining a match region for each host component type found to exist by an agent in the character based user interface (col.5,lines 62-65; col.6,lines 33-36; col.8, lines 27-39); determining whether two or more match regions overlap (col.5,lines 62-65; col.6,lines 33-36; col.8, lines 27-39); and rendering match regions associated with each agent to compose the web enabled user interface (col.8, lines 7-11); an agent manager for determining whether two or more match regions overlap (col.9,lines 23-47; col.6,lines 24-32, 54-64).

Mutschler does not specifically teach recognition and resolution of conflicts between the two or more match regions that overlap; Mutschler further teaches the use of a single process, rather than independent agents. However in the same field of endeavor Galvin teaches recognition and resolution of conflicts between the two or more match regions that overlap (col.5, line 59 – col.6, line 14; when the system is translating from one interface to another there maybe be style inconsistencies, display anomalies, improper overload references; (one skilled in the art would recognize overlapping into one of these categories); Alternatively Galvin does not specifically use the term "overlap" but instead teaches the terms compound GUI objects and overload objects which are related to an overlap of two objects, such that a compound object has more than one element (object). When the compound object is processed from one interface to the web interface the runtime engine takes a compound object and changes it into unary display elements, thus these elements are not "overlapped" compounded together;

(col.11, line 58 – col.12, line 42). Galvin teaches the use of multiple agents when gathering and generating elements from/to interfaces (col.6, lines 15-37).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Galvin into Mutschler, this is true because Galvin is concerned with providing a user a system for translating a conventional application into new GUI (col.1, lines 50-67) much as the same is Mutschler whom is also concerned with providing a user a system for translating a legacy application into a new GUI (web version). One of ordinary skill would recognize Galvin as providing alternative functionality of a module that would detect compound objects and overloaded methods for transforming them to unary display elements such that the elements would not be compounded and rendered to the user for presentation. This process would be recognizable as if two objects were overlapped (such that not all of the objects would be viewable (since they are overlapped/compounded), thus this method makes objects that are not viewable once again viewable by the rendering engine).

As for dependent claims 2, 9 and 16, Mutschler teaches the computer-readable medium of claim 1 wherein the rendering code further comprises: rendering match regions associated with each agent to compose the web enabled user interface (col.9, lines 23-47; col.6, lines 24-32, 54-64).code rendering each match region as a widget, the aggregated widgets composing a formatted output page (col.7, lines 40-65; "window").

Mutschler further teaches the use of a single process, rather than independent agents. However in the same field of endeavor Galvin teaches the use of multiple agents when gathering and generating elements from/to interfaces (col.6, lines 15-37).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Galvin into Mutschler, this is true because Galvin is concerned with providing a user a system for translating a conventional application into new GUI (col.1, lines 50-67) much as the same is Mutschler whom is also concerned with providing a user a system for translating a legacy application into a new GUI (web version). One of ordinary skill would recognize Galvin as providing alternative functionality of a module that would detect compound objects and overloaded methods for transforming them to unary display elements such that the elements would not be compounded and rendered to the user for presentation. This process would be recognizable as if two objects were overlapped (such that not all of the objects would be viewable (since they are overlapped/compounded), thus this method makes objects that are not viewable once again viewable by the rendering engine).

As for dependent claims 3, 11 and 17, Mutschler teaches the computer-readable medium of claim 1 further comprising resolving code executed before the rendering code, comprising: code for resolving a conflict between two or more match regions which overlap based on a policy to determine which agent associated with a match

region controls the overlap region (col.5, lines 62-65; col.6, lines 33-36; col.8, lines 27-39; col.11, line 60 - col.12, line 20).

Mutschler does not specifically teach recognition and resolution of conflicts between the two or more match regions that overlap; Mutschler further teaches the use of a single process, rather than independent agents. However in the same field of endeavor Galvin teaches recognition and resolution of conflicts between the two or more match regions that overlap (col.5, line 59 – col.6, line 14; when the system is translating from one interface to another there maybe be style inconsistencies, display anomalies, improper overload references; (one skilled in the art would recognize overlapping into one of these categories); Alternatively Galvin does not specifically use the term "overlap" but instead teaches the terms compound GUI objects and overload objects which are related to an overlap of two objects, such that a compound object has more than one element (object). When the compound object is processed from one interface to the web interface the runtime engine takes a compound object and changes it into unary display elements, thus these elements are not "overlapped" compounded together; (col.11,line 58 – col.12, line 42). Galvin teaches the use of multiple agents when gathering and generating elements from/to interfaces (col.6, lines 15-37).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Galvin into Mutschler, this is true because Galvin is concerned with providing a user a system for translating a conventional application into new GUI (col.1, lines 50-67) much as the same is Mutschler whom is also concerned with providing a user a system for translating a legacy application into a new GUI (web version). One of

ordinary skill would recognize Galvin as providing alternative functionality of a module that would detect compound objects and overloaded methods for transforming them to unary display elements such that the elements would not be compounded and rendered to the user for presentation. This process would be recognizable as if two objects were overlapped (such that not all of the objects would be viewable (since they are overlapped/compounded), thus this method makes objects that are not viewable once again viewable by the rendering engine).

As for dependent claims 4, 12 and 18, Mutschler teaches the computer-readable medium of claim 3 wherein the resolving code comprises: code for assigning a predetermined priority to each agent; code for comparing the predetermined priority of the two or more conflicting agents; and code for selecting the agent with the highest predetermined priority to control the overlapping region (col.12, lines 28-59).

Mutschler does not specifically teach recognition and resolution of conflicts between the two or more match regions that overlap; Mutschler further teaches the use of a single process, rather than independent agents. However in the same field of endeavor Galvin teaches recognition and resolution of conflicts between the two or more match regions that overlap (col.5, line 59 – col.6, line 14; when the system is translating from one interface to another there maybe be style inconsistencies, display anomalies, improper overload references; (one skilled in the art would recognize overlapping into one of these categories); Alternatively Galvin does not specifically use the term "overlap" but instead teaches the terms compound GUI objects and overload objects which are

related to an overlap of two objects, such that a compound object has more than one element (object). When the compound object is processed from one interface to the web interface the runtime engine takes a compound object and changes it into unary display elements, thus these elements are not "overlapped" compounded together; (col.11, line 58 – col.12, line 42). Galvin teaches the use of multiple agents when gathering and generating elements from/to interfaces (col.6, lines 15-37).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Galvin into Mutschler, this is true because Galvin is concerned with providing a user a system for translating a conventional application into new GUI (col.1, lines 50-67) much as the same is Mutschler whom is also concerned with providing a user a system for translating a legacy application into a new GUI (web version).

As for dependent claims 5, 13 and 19, Mutschler teaches the computer-readable medium of claim 3 wherein the resolving code further comprises: code for comparing the size of the conflicting regions which overlap; and code for selecting the agent having the smaller size region to control the overlapped region (col.12, line 28 – col.13, line 12).

As for dependent claims 7 and 21, Mutschler teaches the computer-readable medium of claim 4 further containing code for controlling the conflicting agents to negotiate whether to relinquish control of at least the overlap region (col.12, lines 28-59; col.9, lines 23-47; col.6, lines 24-32, 54-64).

Mutschler does not specifically teach recognition and resolution of conflicts between the two or more match regions that overlap; Mutschler further teaches the use of a single process, rather than independent agents. However in the same field of endeavor Galvin teaches recognition and resolution of conflicts between the two or more match regions that overlap (col.5, line 59 – col.6, line 14; when the system is translating from one interface to another there maybe be style inconsistencies, display anomalies, improper overload references; (one skilled in the art would recognize overlapping into one of these categories); Alternatively Galvin does not specifically use the term "overlap" but instead teaches the terms compound GUI objects and overload objects which are related to an overlap of two objects, such that a compound object has more than one element (object). When the compound object is processed from one interface to the web interface the runtime engine takes a compound object and changes it into unary display elements, thus these elements are not "overlapped" compounded together; (col.11,line 58 – col.12, line 42). Galvin teaches the use of multiple agents when gathering and generating elements from/to interfaces (col.6, lines 15-37).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Galvin into Mutschler, this is true because Galvin is concerned with providing a user a system for translating a conventional application into new GUI (col.1, lines 50-67) much as the same is Mutschler whom is also concerned with providing a user a system for translating a legacy application into a new GUI (web version).

(Note :) It is noted that any citation to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. In re Heck, 699 F.2d 1331, 1332-33, 216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re Lemelson, 397 F.2d 1006, 1009, 158 USPQ 275, 277 (CCPA 1968)).

Response to Arguments

Applicant's arguments with respect to claims 1-9, 11-21 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Inquires

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nicholas Augustine whose telephone number is 571-270-1056 and fax is 571-270-2056. The examiner can normally be reached on Monday - Friday: 9:30am- 5:00pm Eastern.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Nicholas Augustine/
Examiner
Art Unit 2179
March 25, 2010

/Ba Huynh/
Primary Examiner, Art Unit 2179